



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/717,675	11/21/2000	Christopher G. Kaler	777.340US1	9107

41505 7590 10/19/2005

WOODCOCK WASHBURN LLP  
ONE LIBERTY PLACE - 46TH FLOOR  
PHILADELPHIA, PA 19103

EXAMINER
----------

ALI, SYED J

ART UNIT	PAPER NUMBER
----------	--------------

2195

DATE MAILED: 10/19/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	Application No.	Applicant(s)	
	09/717,675	KALER ET AL.	
	Examiner	Art Unit	
	Syed J. Ali	2195	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

- 1) ☒ Responsive to communication(s) filed on 29 August 2005.
- 2a) ☐ This action is **FINAL**.                      2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

- 4) ☒ Claim(s) 1-12, 14-37, 41-56, 58 and 65 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☒ Claim(s) 58 and 65 is/are allowed.
- 6) ☒ Claim(s) 1-10, 14-31, 33, 34, 36, 37 and 41-56 is/are rejected.
- 7) ☒ Claim(s) 11, 12, 32 and 35 is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \*    c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

- |   |   |
|---|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)             | 4) <input type="checkbox"/> Interview Summary (PTO-413)                     |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)    | Paper No(s)/Mail Date. _____  |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| Paper No(s)/Mail Date _____   | 6) <input type="checkbox"/> Other: _____                                    |

### **DETAILED ACTION**

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on July 18, 2005 has been entered. Claims 1-12, 14-37, 41-56, 58, and 65 are presented for examination.

2. The text of those sections of Title 35, U.S. code not included in this office action can be found in a prior office action.

### ***Claim Objections***

3. **Claim 49 is objected to because of the following informalities:**

a. There is no period at the end of claim 49.

Appropriate correction is required.

### ***Claim Rejections - 35 USC § 102***

4. **Claims 41-48 are rejected under 35 U.S.C. 102(e) as being anticipated by McDaniel (USPN 6,510,214).**

5. As per claim 41, McDaniel teaches the invention as claimed, including a method in a computer system for servicing requests from multiple client computers, the method comprising:

Art Unit: 2195

monitoring a quantity of work being performed by the computer system (col. 3 lines 46-67);

determining whether the quantity has exceeded an upper limit (col. 4 lines 7-9); and

if the quantity has exceeded the upper limit but has not dropped below a lower limit (col. 4 lines 9-11), not accepting new requests into the computer system (col. 4 lines 1-4), wherein the upper and lower limits are not equal (col. 4 lines 7-11).

6. As per claim 42, McDaniel teaches the invention as claimed, including the method as claimed in claim 41, further comprising, if the quantity has exceeded the upper limit and has dropped below the lower limit, accepting the new requests into the computer system (col. 4 lines 9-11).

7. As per claim 43, McDaniel teaches the invention as claimed, including the method as claimed in claim 41, further comprising continuing to monitor the quantity of work being performed by the computer system after the quantity has exceeded the upper limit (col. 3 lines 46-67).

8. As per claim 44, McDaniel teaches the invention as claimed, including the method as claimed in claim 41, wherein the quantity of work is indicated by a number of work items on one or more work queues (col. 3 lines 54-58), and determining whether the quantity has exceeded the upper limit comprises determining whether the number of work items has exceeded a predefined value (col. 3 lines 60-62).

9. As per claims 45-47, McDaniel teaches the invention as claimed, including a method of servicing requests from multiple client computers as described in claims 41-43, wherein the quantity of work is defined as an amount of time to return a result (col. 3 lines 46-54).

10. As per claim 48, McDaniel teaches the invention as claimed, including the method as claimed in claim 45, wherein the amount of time includes a time it takes to post the result to an output port (col. 3 lines 31-40, 46-54), and determining whether the amount of time has exceeded the upper limit comprises determining whether the time it takes to post the result has exceeded a predefined value (col. 3 lines 40-45, 60-62).

***Claim Rejections - 35 USC § 103***

11. **Claims 1-10 and 14-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shah et al. (USPN 6,226,689) (hereinafter Shah) in view of Sager (USPN 6,542,921).**

12. As per claim 1, Shah teaches the invention as claimed, including a method in a computer system for servicing requests from one or more client computers, the method comprising:

receiving a request from a client computer (col. 3 lines 51-54);

a first thread processing the request by invoking a receive handler that creates a work item, wherein the first thread is part of a pool of generic threads (col. 3 lines 51-54);

a second thread performing a task specified in the work item by invoking a work handler, wherein the second thread is part of the pool of generic threads (col. 4 lines 58-67);

Art Unit: 2195

receiving a result of performing the task (col. 3 lines 54-57); and

a third thread returning at least a portion of the result to the client computer by invoking a reply handler, wherein the third thread is part of the pool of generic threads (col. 3 lines 54-57).

13. Sager teaches the invention as claimed, wherein if a quantum has not expired for the first thread, the second thread or the third thread, then the first thread, the second thread or the third thread being given an additional work item to perform without relinquishing the central processing unit upon which the first thread, the second thread or the third thread was running (col. 9 lines 2-7, 30-32, 55-67; col. 11 lines 4-29).

14. It would have been obvious to one of ordinary skill in the art to combine Shah and Sager to avoid bottlenecked processing. In particular, where one thread is performing work that is necessary for completion of a second thread, it is imperative that the first thread completes its work in a timely fashion. This is commonly referred to as "livelock", a condition that Sager seeks to remedy. A first thread is guaranteed a particular share of a resource such that it has progressed through its workload sufficiently for the second thread to make full use of its time slice.

15. As per claim 2, Shah teaches the invention as claimed, including the method as claimed in claim 1, wherein receiving the request comprises:

receiving the request into an input/output port (col. 6 lines 36-41); and

placing a reference in a queue indicating that work is available for the first thread (col. 6 lines 59-65).

Art Unit: 2195

16. As per claim 3, Shah teaches the invention as claimed, including the method as claimed in claim 2, further comprising:

invoking a receive thread manager when the work is available (col. 6 lines 46-50); and  
the receive thread manager scheduling the first thread for execution on one of multiple processors (col. 4 lines 58-67).

17. As per claim 4, Shah teaches the invention as claimed, including the method as claimed in claim 3, wherein the receive thread manager schedules the first thread from a queue of available threads (col. 6 lines 46-67).

18. As per claim 5, Shah teaches the invention as claimed, including the method as claimed in claim 1, further comprising the first thread placing the work item on a work queue for execution by the second thread (col. 6 lines 59-65).

19. As per claim 6, Shah teaches the invention as claimed, including the method as claimed in claim 5, further comprising the first thread placing a reference in a completion port queue, indicating that work is available for the second thread (col. 6 lines 36-41).

20. As per claim 7, Shah teaches the invention as claimed, including the method as claimed in claim 1, further comprising the second thread creating and placing a second work item on a reply work queue when the results are received (col. 7 lines 17-27).

Art Unit: 2195

21. As per claim 8, Shah teaches the invention as claimed, including the method as claimed in claim 7, further comprising the second thread placing a reference in a completion port queue, indicating that work is available for the third thread (col. 6 lines 36-41; col. 7 lines 17-27).

22. As per claim 9, Shah teaches the invention as claimed, including the method as claimed in claim 1, further comprising:

receiving input data (col. 6 lines 1-45);

the first thread storing the input data in a cache that is accessible to the second thread (col. 6 lines 1-45).

23. As per claim 10, Shah teaches the invention as claimed, including the method as claimed in claim 1, wherein the results include data, and further comprising storing the data in a cache that is accessible to the third thread (col. 6 lines 1-45).

24. As per claim 14, Shah teaches the invention as claimed, including a method in a computer system for servicing requests from multiple client computers, the method comprising:

receiving a request from a client computer to perform a function (col. 3 lines 51-54);

performing a first task, by a first work handler invoked by a first thread in a ready state, wherein the first task is associated with a first state of the function (col. 3 lines 51-54), performing the first task includes issuing an asynchronous request for data (col. 9 lines 28-30);

placing the first thread back in the ready state (col. 6 lines 65-67);

receiving the data specified in the asynchronous request (col. 11 lines 34-37); and



performing a second task, by a second work handler invoked by a second thread in the ready state, wherein the second task is associated with a second state of the function, and the second task performs an operation on the data (col. 4 lines 58-67), wherein the first thread and the second thread are all identical generic threads within a pool of threads (col. 1 lines 14-21).

25. Sager teaches the invention as claimed, including the request being a multi-state function (col. 2 lines 27-37; col. 9 lines 2-7, 30-32, 55-67). Sager teaches multiple worker threads cooperating to perform a task, as claimed. The “multi-state” function essentially processes part of a task in one thread, prepares work for a second thread, wherein the second thread takes over processing and performs additional work. This is the same way that multithreading is employed by Sager, with additional features to prevent livelock, wherein the second thread has no work to perform due to the first thread inefficiently handling its own workload.

26. As per claim 15, Shah teaches the invention as claimed, including the method as claimed in claim 14, further comprising:

processing the request by a receive handler invoked by a third thread (col. 3 lines 54-57);  
creating a work item that specifies the first task (col. 4 lines 58-67); and  
placing the work item in a work queue that is accessible to the first thread (col. 6 lines 36-41).

27. As per claim 16, Shah teaches the invention as claimed, including the method as claimed in claim 15, wherein the first thread, the second thread, and the third thread are all identical generic threads within the pool of generic threads (col. 3 lines 57-61).

28. As per claim 17, Sager teaches the invention as claimed, including the method as claimed in claim 15, wherein placing the work item in the work queue comprises placing the work item in a low priority work queue, the method further comprising:

placing a second work item that specifies the second task on a high priority work queue (col. 9 lines 41-51);

a work handler looking for a work item first on the high priority work queue (col. 19 lines 18-22); and

if no work item exists on the high priority work queue, the work handler looking for the work item on the low priority queue (col. 9 lines 41-51; col. 10 lines 55-61).

29. As per claim 18, Shah teaches the invention as claimed, including the method as claimed in claim 14, wherein the asynchronous request is a request that would otherwise cause the first thread to block (col. 9 lines 28-30).

30. As per claim 19, Shah teaches the invention as claimed, including the method as claimed in claim 14, wherein the asynchronous request is a request that would otherwise cause a blocking condition to occur (col. 9 lines 28-30).

31. As per claim 20, Shah teaches the invention as claimed, including the method as claimed in claim 14, wherein issuing the asynchronous request comprises issuing the asynchronous request to a database manager (col. 9 lines 28-30; col. 11 lines 35-38).

32. As per claim 21, Shah teaches the invention as claimed, including the method as claimed in claim 20, further comprising:

the database manager placing the asynchronous request on a pending queue (col. 9 lines 28-30; col. 11 lines 35-38; col. 6 lines 59-65); and

when the data is received, placing a work item associated with the second state on a work queue (col. 6 lines 59-65).

33. As per claim 22, Sager teaches the invention as claimed, including the method as claimed in claim 21, wherein the work queue is a high priority work queue, the method further comprising:

placing a first work item that specifies the first task on a low priority work queue (col. 9 lines 41-51);

a work handler looking for a work item first on the high priority work queue (col. 19 lines 18-22); and

if no work item exists on the high priority work queue, the work handler looking for the work item on the low priority work queue (col. 9 lines 41-51; col. 10 lines 55-61).

34. As per claim 23, Shah teaches the invention as claimed, including the method as claimed in claim 14, further comprising returning a result of the first task and the second task by invoking third thread, which in turn invokes a reply handler to return the result to the client computer (col. 3 lines 54-57).

35. As per claim 24, Shah teaches the invention as claimed, including the method as claimed in claim 23, wherein the first thread, the second thread, and the third thread are all identical generic threads within a pool of generic threads (col. 3 lines 57-61).

36. As per claim 25, Shah teaches the invention as claimed, including the method as claimed in claim 14, further comprising:

performing additional tasks associated with subsequent function states by additional work handlers invoked by one or more additional threads (col. 6 lines 65-67);

at least some of the additional work handlers issuing additional requests (col. 7 lines 17-27); and

placing threads associated with the at least some of the additional work handlers back in the ready state after issuing the additional requests (col. 6 lines 65-67); and

the requests being asynchronous requests (col. 9 lines 28-30; col. 11 lines 35-38).

37. As per claim 26, Shah teaches the invention as claimed, including a method in a computer system for servicing requests from multiple client computers, the method comprising:

determining that work is available after receiving a request from a client computer (col. 3 lines 54-57; col. 4 lines 58-67); and

when work is available, a first work handler invoked by a first thread looking in a first work queue for a first work item corresponding to the work (col. 4 lines 58-67), wherein the first thread is a generic thread within a pool of generic threads (col. 1 lines 14-21).

Art Unit: 2195

38. Sager teaches the invention as claimed, including the request being a multi-state function (col. 2 lines 27-37; col. 9 lines 2-7, 30-32, 55-67); and

if the first work item is not found in the first work queue, the first work handler looking in a second work queue for the first work item (col. 9 lines 41-51; col. 10 lines 55-61; col. 19 lines 18-22).

39. As per claim 27, Shah teaches the invention as claimed, including the method as claimed in claim 26, further comprising:

receiving a request from a client computer to perform a task (col. 3 lines 54-57);

creating the first work item that specifies the task (col. 3 lines 54-57; col. 4 lines 58-67);

placing the first work item in the second queue (col. 3 lines 54-57; col. 4 lines 58-67);

and

indicating that the work is available (col. 4 lines 58-67).

40. As per claim 28, Shah teaches the invention as claimed, including the method as claimed in claim 26, further comprising:

the first work handler performing a task specified in the first work item (col. 4 lines 58-67); and

issuing an asynchronous request for data (col. 9 lines 28-30).

41. As per claim 29, Shah teaches the invention as claimed, including the method as claimed in claim 28, further comprising:

Art Unit: 2195

receiving the data (col. 6 lines 1-45);  
placing a second work item on the first work queue (col. 4 lines 58-67; col. 6 lines 1-45);  
and  
indicating that additional work is available (col. 4 lines 58-67; col. 6 lines 1-45).

42. As per claim 30, Sager teaches the invention as claimed, including the method as claimed in claim 29, further comprising:

a second work handler invoked by a second thread looking in the first work queue for the second work item when the second work is available (col. 9 lines 41-51; col. 10 lines 55-61; col. 19 lines 18-22), wherein the first and second threads are all identical threads within the pool of generic threads (col. 9 lines 41-51); and

performing a second task specified in the second work item (col. 9 lines 41-51; col. 10 lines 55-61; col. 19 lines 18-22).

43. As per claim 31, Sager teaches the invention as claimed, including the method as claimed in claim 26, wherein the computer system includes multiple work queues, including the first work queue and the second work queue, each of the multiple work queues are associated with a priority level, and wherein the method further comprises:

looking for the first work item first in a work queue associated with a highest priority level (col. 9 lines 41-51; col. 10 lines 55-61; col. 19 lines 18-22); and

if the first work item is not found in the work queue associated with the highest priority level, looking for the first work item in each of the multiple work queues in descending priority

Art Unit: 2195

order until the first work item is found (col. 9 lines 41-51; col. 10 lines 55-61; col. 19 lines 18-22).

**44. Claims 33-34, 36-37, 49-52 and 55-56 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shah in view of Wolrich et al. (USPN 6,625,654) (hereinafter Wolrich).**

45. As per claim 33, Shah teaches the invention as claimed, including a method in a computer system for servicing requests from multiple client computers, the method comprising:

receiving, from a client computer, a request to perform a first task (col. 3 lines 51-54), wherein the task is handled by threads that are all identical threads within a pool of generic threads (col. 1 lines 14-21).

46. Wolrich teaches the invention as claimed, including each thread being given a specialized function, such that particular tasks are allocated to threads that are optimized for performance of that task (col. 9 lines 48-56; col. 10 lines 8-17).

47. Though Wolrich does not specifically address complex logic or long-running tasks, it would have been obvious to one of ordinary skill in the art that specialized threads can be developed for any type of task. Wolrich merely provides some examples of the types of tasks that would benefit from such optimization; the list is not meant to be exhaustive. Furthermore, it would have been obvious to one of ordinary skill in the art to combine Shah and Wolrich, since compute-intensive or other tasks that have higher resource demands stand to benefit greatly from software that is optimized for performance of those tasks.

Art Unit: 2195

48. As per claim 34, Sager teaches the invention as claimed, including the method as claimed in claim 33, wherein the first thread is a thread within a first group of threads having a first priority level, and the second thread is a thread within a second group of threads having a second priority level that is lower than the first priority level (col. 9 lines 41-51; col. 10 lines 55-61; col. 19 lines 18-22).

49. As per claim 36, Shah teaches the invention as claimed, including the method as claimed in claim 33, further comprising:

the second handler returning a result of the first task (col. 3 lines 54-57); and  
using the result, performing a second task by a third handler invoked by a third thread of the first group of threads (col. 3 lines 54-57).

50. As per claim 37, Sager teaches the invention as claimed, including the method as claimed in claim 36, wherein the first task is specified in a first work item and the second task is specified in a second work item, the method further comprising:

the first handler obtaining the first work item from a first work queue (col. 9 lines 41-51; col. 10 lines 55-61; col. 19 lines 18-22); and  
the third handler obtaining the second work item from a second work queue (col. 9 lines 41-51; col. 10 lines 55-61; col. 19 lines 18-22).

51. As per claim 49, Shah teaches the invention as claimed, including an application program for implementation by an application server, the application program comprising:



at least one receive handler that can be invoked by a thread within a pool of threads (col. 3 lines 51-54), wherein each thread in the pool of threads is identical (col. 3 lines 57-61);

at least one work handler that also can be invoked by the thread (col. 4 lines 58-67); and

at least one reply handler that also can be invoked by the thread (col. 3 lines 54-57).

52. Wolrich teaches the invention as claimed, including each thread being given a specialized function, such that particular tasks are allocated to threads that are optimized for performance of that task (col. 9 lines 48-56; col. 10 lines 8-17), e.g. complex or long-running logic tasks.

53. As per claim 50, Shah teaches the invention as claimed, including the application program as claimed in claim 49, wherein a receive handler of the at least one receiver handler is executed when the application server receives a request from a client computer (col. 3 lines 54-57), and the receiver handler creates a first work item to be performed by a work handler of the at least one work handler (col. 3 lines 51-54).

54. As per claim 51, Shah teaches the invention as claimed, including the application program as claimed in claim 50, wherein the work handler is executed when the first work item exists (col. 6 lines 46-50), the work handler receives results (col. 3 lines 54-57), and the work handler creates a second work item to be performed by a reply handler of the at least one reply handler (col. 6 lines 36-41; col. 7 lines 17-27).

55. As per claim 52, Shah teaches the invention as claimed, including the application program as claimed in claim 51, wherein the reply handler is executed when the second work

Art Unit: 2195

item exists, and the reply handler sends the results to the client computer (col. 3 lines 54-57; col. 6 lines 36-41; col. 7 lines 17-27).

56. As per claim 55, Shah teaches the invention as claimed, including the application program as claimed in claim 49, wherein multiple identical copies of the thread exist within the pool of threads, and at least one copy of the thread is executed each time a request is received from a client computer (col. 3 lines 48-61).

57. As per claim 56, Shah teaches the invention as claimed, including the application program as claimed in claim 49, wherein multiple copies of the thread can simultaneously be executed by multiple processors available to the application server (col. 3 lines 18-24; col. 3 lines 48-61).

58. **Claims 53-54 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shah in view of Wolrich and further in view of Sager.**

59. As per claim 53, Shah teaches the invention as claimed, including the application program as claimed in claim 49, wherein the thread can invoke multiple work handlers (col. 4 lines 58-67); and

at least some of the multiple work handlers issue asynchronous requests for data (col. 9 lines 28-30).

60. Neither Shah nor Wolrich discuss “multi-state” functions, such that threads progress from a working state to a ready state.

61. Sager teaches the features of the claims not discussed by Shah or Wolrich, i.e. where some of the multiple work handlers are designed to perform tasks associated with various states of a multi-state function, wherein the thread is then placed back in a ready state to execute a subsequent work item (col. 2 lines 27-37; col. 9 lines 2-7, 30-32, 55-67).

62. It would have been obvious to one of ordinary skill in the art to combine Shah and Wolrich with Sager to avoid bottlenecked processing. In particular, where one thread is performing work that is necessary for completion of a second thread, it is imperative that the first thread completes its work in a timely fashion. This is commonly referred to as “livelock”, a condition that Sager seeks to remedy. A first thread is guaranteed a particular share of a resource such that it has progressed through its workload sufficiently for the second thread to make full use of its time slice.

63. As per claim 54, Shah teaches the invention as claimed, including the application program as claimed in claim 53, wherein when the data is returned, a second work handler is executed (col. 3 lines 54-57).

*Allowable Subject Matter*

64. **Claims 11-12, 32, and 35 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.**

65. **Claims 58 and 65 are allowed.**

*Response to Arguments*

66. **Applicant's arguments with respect to claims 1-10, 14-31, 33-34, 36-37, and 41-56 have been considered but are moot in view of the new grounds of rejection.**

*Conclusion*

67. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Syed J. Ali whose telephone number is (571) 272-3769. The examiner can normally be reached on Mon-Fri 8-5:30, 2nd Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai T. An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



Syed Ali  
October 11, 2005



MENG-AI T. AN  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100